

Les commandes

Sommaire

[Concept](#)
[Onglet Général](#)
[Onglet Expert](#)

Concept

Les commandes sont des éléments de configuration permettant de décrire l'utilisation d'un plugin.

Elle précise où trouver le plugin, comment l'utiliser avec les bons arguments et comment remplacer ces arguments par des données.

Afin d'avoir des commandes génériques, il convient d'utiliser le mécanisme des variables pour accéder aux données (voir la page [Les Variables \(Remplacement dynamique de contenu - Anciennement les Macros \)](#)).

Onglet Général

	Propriété	Valeur
Expert	Nom de la Commande *	
	Ligne de Commande *	
	Temps maximum d'exécution d'une commande (secondes)	Par défaut [Même comportement que son check]
	Seuil d'alerte de l'utilisation CPU (secondes)	Par défaut [Même comportement que son check]
	Activé	<input type="radio"/> Vrai <input type="radio"/> Faux <input checked="" type="radio"/> Par défaut [Vrai]
	Pack	

Nom de la commande

- *Clé d'import:* `command_name`
Cette propriété est utilisée pour définir un nom court permettant d'identifier une commande.

Ligne de commande

- *Clé d'import:* `command_line`
Cette propriété est utilisée pour définir la commande à lancer. Si vous utilisez des Variables, leurs noms doivent être en majuscules. Si elles sont écrites en minuscules, les Variables seront donc transformées en majuscules dans la ligne de commande.

Temps maximum d'exécution d'une commande (secondes)

- *Clé d'import:* `timeout`
Cette propriété est utilisée pour définir le temps maximum d'exécution d'une commande. Si paramétrée à -1, le temps maximum d'exécution d'une commande sera récupéré, dans l'ordre, par l'élément Check, par l'élément Hôte, ou par la valeur par défaut (`shinken.cfg: check_running_timeout`).
 La limite est fixée à 157852800 (soit 5 ans).

Seuil d'alerte de l'utilisation CPU (secondes)

- *Clé d'import:* warning_threshold_cpu_usage
Nombre de secondes que peut consommer une commande avant d'apparaître en Warning dans le check 'Scheduler - Performance'
La valeur sera récupérée dans l'ordre, par la commande, par l'élément Check, par l'élément Hôte, ou par la valeur par défaut (shinken.cfg: warning_threshold_cpu_usage)
[i La limite est fixée à 157852800 \(soit 5 ans\).](#)

Activé

- *Clé d'import:* enabled
Cette propriété est utilisée pour définir si la commande est activée ou non dans la configuration.

Pack

- *Clé d'import:* pack
Cette propriété est utilisée pour définir si la commande est activée ou non dans la configuration.

Onglet Expert

Propriété	Valeur
Tag Poller	-- Par défaut [Même comportement que son parent (hôte, cluster ou check)] --
Tag de Reactionner	-- Par défaut [Même comportement que son parent (hôte, cluster ou check)] --
Type de Module	Par défaut [Fork]
Exécuté dans un Shell	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux <input checked="" type="checkbox"/> Par défaut [Faux]

Tag Poller

- *Clé d'import:* poller_tag
Cette propriété est utilisée pour définir dans quels Pollers la commande devrait être exécutée.
Par défaut, la valeur de Tag de Poller est "Même comportement que son parent (hôte, cluster ou check)".
Si la valeur est "non tagué" :
 - tous les Pollers n'ayant aucun poller_tag seront utilisés pour l'exécution
 - ou tous les Pollers spécifiant qu'ils prennent aussi les checks non tagués

Tag Reactionner

- *Clé d'import:* reactionner_tag
Cette propriété permet de définir quels Reactionners vont exécuter cette commande, lorsque celle-ci est utilisée en commande de notifications ou dans le gestionnaire d'événements.

Type de module

- *Clé d'import:* module_type
Cette propriété est utilisée pour définir si cette commande devrait être exécutée par un module spécifique de vos pollers ou reactionners.

Exécuté dans un Shell

- *Clé d'import:* shell_execution
Cette propriété est utilisée pour définir si la commande doit être exécutée dans un shell.
 - Vrai:
 - La commande sera exécutée dans un shell (/bin/sh) et vous pourrez donc utiliser les fonctionnalités spéciales du shell comme | ; ou &&.



Si vos utilisateurs peuvent changer les paramètres de commandes, cela peut être un **risque de sécurité** qui résultera en une exécution shell d'une commande utilisateur.

○ Faux:

- La commande ne sera pas exécutée dans un shell et donc vous **ne pourrez pas** utiliser les fonctionnalités spéciales du shell comme `|` ; ou `&&` mais votre sécurité sera augmentée car vous serez certain que les utilisateurs ne pourront pas exploiter vos commandes pour lancer une autre commande shell.